

Learning measures of semi-additive behaviour

Hamidreza Chinaei^{*}
Department of Computer
Science
University of Toronto
Toronto, ON, Canada
chinaei@cs.toronto.edu

Mohsen Rais-Ghasem
Business Analytics Team
IBM Canada
Ottawa, ON, Canada
mohsen.rais-
ghasem@ca.ibm.com

Frank Rudzicz
Department of Computer
Science
University of Toronto
Toronto, ON, Canada
frank@cs.toronto.edu

ABSTRACT

In business analytics, *measure* values, such as sales numbers or volumes of cargo transported, are often summed along values of one or more corresponding *categories*, such as time or shipping container. However, not every measure should be added by default (e.g., one might more typically want a *mean* over the heights of a set of people); similarly, some measures should only be summed within certain constraints (e.g., population measures need not be summed over years). In systems such as Watson Analytics, the exact additive behaviour of a measure is often determined by a human expert. In this work, we propose a small set of features for this issue. We use these features in a case-based reasoning approach, where the system suggests an aggregation behaviour, with 86% accuracy in our collected dataset.

1. INTRODUCTION

In business analytics, *measure* values are often summed, but often other aggregation measures, such as means or variances, is more appropriate. For example, the dataset in Figure 1 contains employment statistics for Australian states for 2007 and 2008; clearly, while summing over partitions by year may be appropriate, but summing over the entire column is not. The result of a default aggregation (sum) over these data is shown in Figure 2 given the user question: "What are the values of Total Fully Employed by State?". Figure 3 shows another example dataset that tracks information about loan requests received by branches of a bank. The number of clients is incorrectly added in a branch by branch analysis (see Figure 4). Here, adding the loan amounts for each branch makes sense, but adding the number of clients does not. This problem is usually avoided

^{*}Work done while at IBM Canada.

Year	State	Total Population	Total Fully Employed
2007	Australian Capital Territory	344,176	147,534
2007	New South Wales	6,883,852	2,392,606
2007	Northern Territory	216,618	85,902
2007	Queensland	4,159,990	1,541,242
2007	South Australia	1,578,489	523,037
2007	Tasmania	495,858	156,584
2007	Victoria	5,199,503	1,838,802
2007	Western Australia	2,135,006	797,062
2008	Australian Capital Territory	351,101	149,031
2008	New South Wales	7,001,782	2,459,047
2008	Northern Territory	222,526	88,663
2008	Queensland	4,275,551	1,577,158
2008	South Australia	1,597,880	540,869
2008	Tasmania	501,774	164,476
2008	Victoria	5,313,285	1,856,459
2008	Western Australia	2,208,928	848,651

Figure 1: The snapshot of a spreadsheet containing statistics for employment in Australian states.



Figure 2: The result of a default aggregation on the spreadsheet of statistics for employment in Australian states for the user question: "What are the values of Total Fully Employed by State?".

by defining the correct default aggregation behaviour ahead of time, through data modeling techniques, which often requires hand-tuning each variable in a dataset.

In this work, we propose an alternative approach. We introduce a small set of features and develop them in a CBR system to suggest better default aggregations and hence

A	B	C	D	E
Branch	Number of custom	Customer ID	Loan Amount (x1000)	
3	3,017	10,012	2.99	
3	3,017	10,017	5.05	
3	3,017	10,030	1.04	
3	3,017	10,039	1.75	
3	3,017	10,069	0.76	
3	3,017	10,071	1.46	
3	3,017	10,096	0.95	
3	3,017	10,128	1.05	
3	3,017	10,129	3.36	
3	3,017	10,140	0.31	
3	3,017	10,142	6.69	
3	3,017	10,169	2.39	
3	3,017	10,197	9.52	
3	3,017	10,200	0.17	
3	3,017	10,218	2.33	
3	3,017	10,234	0.83	
3	3,017	10,351	1.84	

Figure 3: A snapshot of the bank loan spreadsheet.



Figure 4: The result of a default aggregation on the bank loan spreadsheet, for the user’s question: “Number of customers for each branch?”.

eliminate or at least accelerate prior, data-specific modeling. This is done by extracting potentially useful features from the data and by applying case-based reasoning (CBR) [Riesbeck and Schank, 1989; Kolodner, 1993].

Our CBR system is fed a few use cases representing an aggregation context (so called *known cases*), consisting of (i) a *measure* item whose default aggregation is being learned or queried (e.g. “Total Fully Employed” in Figure 1); (ii) one or more *category* items for which the given measure has repeated values that need to be aggregated (e.g. “State” and “Year” in Figure 1); (iii) the expected default aggregate which is one of *sum*, *average*, or *last-period*.

In this work, we propose a small set of features that, when used in our CBR system, improve the learning of appropriate aggregate actions up to 86% accuracy. In the rest of this paper, we first briefly explain CBR and our CBR system architecture in Section 2. We describe our extracted features for our CBR package in Section 3. The similarity measure for each feature is described in Section 4. We then go through our empirical settings in Section 5. Finally, we

conclude and address future directions in Section 6.

2. CBR ARCHITECTURE

CBR has been applied successfully in many practical domains [Lamontagne and Plaza, 2014; Jalali and Leake, 2014; Chen et al., 2014; Dong et al., 2014; Freyne and Smyth, 2010; Jurisica, 1997]. The principal idea in CBR is to draw parallels between a new case and those that have been already solved.

Figure 5 shows an example case represented by feature values that need to be extracted. The action for each known case is defined and learned. The best action for the current case is selected using the most similar cases to the current one. That is, the current case is compared to the known cases using feature similarity measures and the case features, then the action for the most similar case, among all known cases, is selected for and applied to the current case.

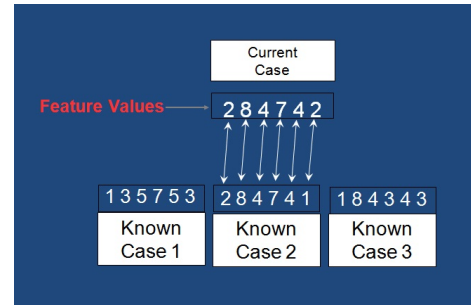


Figure 5: Case-based reasoning (CBR): In CBR a new case (current case) is compared against existing cases using a set of extracted features.

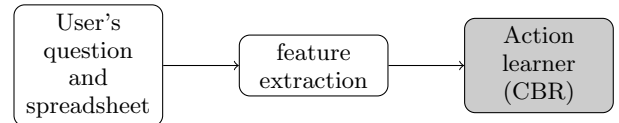


Figure 6: The architecture of our CBR system.

Figure 6 demonstrates the architecture of our developed aggregate learner. The user’s spreadsheet is submitted to the system followed by the user’s question(s). The system extracts the features for the spreadsheet particularly for *category* and *measure* columns. These features are described in Section 3. The action learner component receives the features and finds the aggregate action for each measure column based on CBR.

3. EXTRACTED FEATURES

At the heart of our system, and any CBR system, lies a carefully selected *feature set* that is used to measure the similarity of the current case to known cases, as depicted in Figure 5. The role that each feature plays in the overall similarity measurement (i.e., the feature weights) can be assigned manually or automatically using a number of optimization techniques similar to those of Lamontagne and Guyard [2014].

Table 1 demonstrates the representation of a case using the extracted features for the bank loan example in Figure 3. Our feature set consists of factors that we found most use-

Table 1: Example of a case representation in our CBR system.

dataset:	bankloan.xls
measure column:	“Loan Amount (x1000)”
category column:	“Branch”
concepts (measure column):	[metric, monetary]
concepts (category column):	[attribute]
association type (category to measure):	one-to-many
averaged CoV:	1.37
measure aggregate action:	sum

ful in establishing the default aggregation behaviour, most notably the following features:

1. *Semantic annotations* of measure columns and category columns, similar to column *concepts* as defined in Rais-Ghasem et al. [2013];
2. *Association type* between category columns and measure columns;
3. *Coefficient of variation (CoV)* as an indication of trends of measure values in the context of given categories.

Semantic annotations

For the semantic annotations of measures and categories (feature 1), consider the measure column “Loan Amount (x1000)”, and the category column “Branch”. The semantic annotations of the measure column are captured as *metric* and *monetary*, and the semantics annotations of the category column is captured as *attribute* [Rais-Ghasem et al., 2013]. In particular, we used a designation shared by many metrics in our known cases in our CBR system, and specifically avoided depending on explicit semantic knowledge about items.

Association types

A simple analysis of data values in Figure 3 reveals that values of column “A” (“Branch”) and “B” (“Customer ID”) have a *one-to-one association type*, whereas the values of column “A” and “D” (Loan Amount (x1000)) have a *one-to-many association type*. That is, for each “Branch” number there is only one value for “number of customer”, thus a *one-to-one* association type. Note that the association type feature between a category column and measure column can be assigned four different values: *one-to-one*, *one-to-many*, *many-to-one*, and *many-to-many*.

As useful as this might be (i.e., it may suggest a non-additive behaviour for *B* in the context of *A*), it is not sufficient to identify the correct behaviour for the employment use case in Figure 1. That is, for each category column “Year” (or “State”) there is many “Total Fully Employed” values, thus *many-to-many* association type. However, addition is not the correct aggregate behaviour for “Total Fully Employed” over “Year”. We thus introduce the third feature that can potentially solve this limitation.

Value trends

We observe that trends of value dynamics are often useful, especially in combination with the aforementioned association type and semantic annotation of category (e.g., “Year” is a temporal category, whereas “Branch” is not).

To quantify data trends, we use the *coefficient of variation (CoV)*:

$$CoV = \sigma/\mu$$

where σ is the standard deviation and μ is the mean. Here, *CoV* acts as normalized standard variation. The lower the *CoV* measure for a column of data, the greater the trend tended to be in early empirical tests. This can be seen in Figure 7 to Figure 10.

Figure 7 shows the *CoV* for the employment status. The figure shows that the *CoV* for “Totally Fully Employed” over “Year” is close to 0. Note that the average of *CoV* over different states is 0.01.

We have done a similar analysis for data in Watson Analytics that contains temperatures for Canadian cities. Figure 8 shows a snapshot of this spreadsheet. Figure 9 shows the *CoV* for the temperature, averaged over the category column (cities). This figure shows that the *CoV* for “Mean Temperature” over “Day of Year” is between 0 and 1. That is, the average *CoV* over different cities is 0.48. In this analysis, to get sensible *CoV* values we had to transform the measure values to positive (a negative measure value occurs in the column). To do so, we add the absolute value of the minimum value of the measure column to all values in that measure column.

City	Year	Month	Day	Day of Year	Day of Week	Mean Temp
Ottawa	2005	1	1	1	Saturday	-3.3
Ottawa	2005	1	2	2	Sunday	-10
Ottawa	2005	1	3	3	Monday	-2.9
Ottawa	2005	1	4	4	Tuesday	-3.9
Ottawa	2005	1	5	5	Wednesday	-10.6
Ottawa	2005	1	6	6	Thursday	-12.5
Ottawa	2005	1	7	7	Friday	-7.4
Ottawa	2005	1	8	8	Saturday	-4
Ottawa	2005	1	9	9	Sunday	-5.5
Ottawa	2005	1	10	10	Monday	-3.8
Ottawa	2005	1	11	11	Tuesday	-12.8
Ottawa	2005	1	12	12	Wednesday	-6.7
Ottawa	2005	1	13	13	Thursday	5.1
Ottawa	2005	1	14	14	Friday	-4.7
Ottawa	2005	1	15	15	Saturday	-9.4
Ottawa	2005	1	16	16	Sunday	-11.5

Figure 8: The temperature readings spreadsheet.

Finally, we follow a similar analysis over bank loan data, whose *CoV* is shown in Figure 10. Here, *CoV* for “Loan Amount” over “Customer ID” is close to 1 and the average *CoV* over different branches is 1.14.

Thus an interesting pattern emerges, as summarized in Table 2. *CoV* for measure values repeated for a category (e.g., various employment figures for different years) had strong associations with expected aggregation functions. Using *CoV* in combination with other features to propose default aggregation automatically, in general, within IBM Wat-

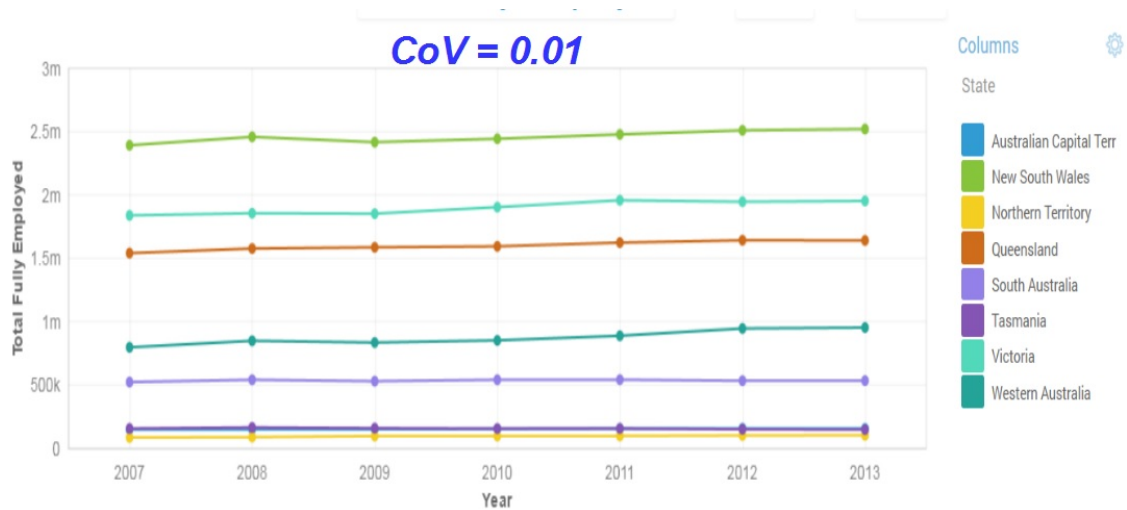


Figure 7: The trend of Australia states data. The CoV for “Totally Fully Employed” over “Year” is close to 0.

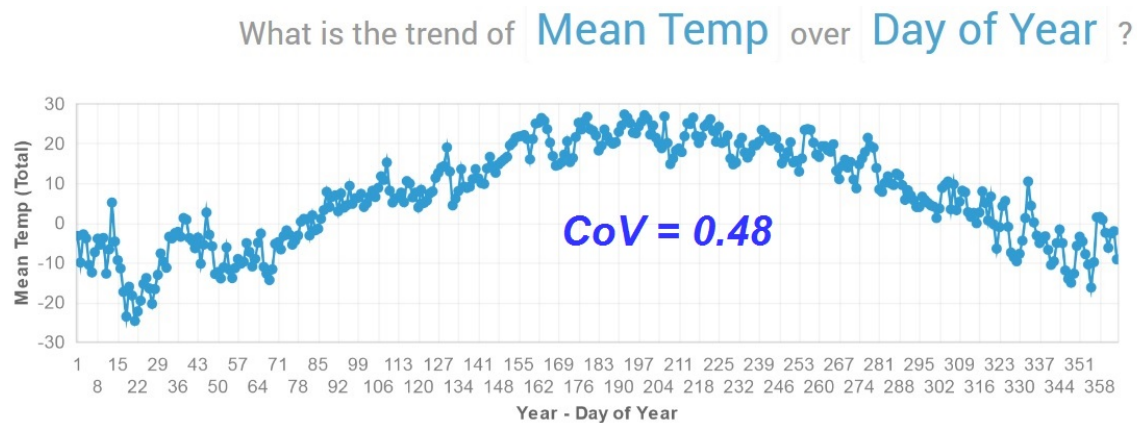


Figure 9: The trend of temperature data. The CoV for “Mean Temp” over “Days of Year” is between 0 and 1.

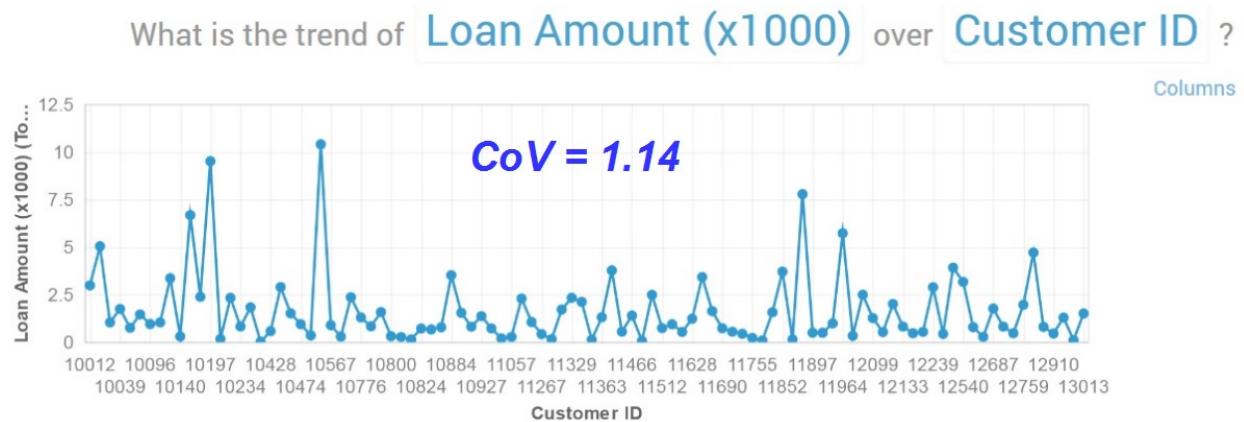


Figure 10: The trend of bank loan data. The CoV for “Loan Amount” over “Customers” is close to 1.

son Analytics.

Table 2: The *CoV* patterns for aggregate learning.

Measure <i>CoV</i> values	Default Aggregation
Values close to 0	<i>last-period</i>
Values close to 0.5	<i>average</i>
Values close to 1	<i>sum</i>

4. FEATURE SIMILARITY FUNCTION

The similarity function for each feature is defined as follows.

Feature 1 (column concepts):

$$Sim(f_1(c_1), f_1(c_2)) = \frac{card(f_1(c_1) \cap f_1(c_2)) * 2}{card(f_1(c_1)) + card(f_1(c_2))}$$

where c_1 and c_2 are case 1 and case 2, respectively; and $f_1(c_i)$ returns the value of feature one for case i . Note that feature 1 is the list of column concepts. In the above formula, $card()$ returns the cardinality of each set. Effectively, the similarity is the number of common concepts (feature 1 values) for case 1 and case 2 divided by the total number of column concepts. Clearly, if the two cases share no concept, the similarity is 0, and if the two cases are identical, the similarity is 1.

Feature 2 (association type between columns): This similarity measure is defined by our domain experts as follows: If the association types are the same for the two cases then similarity is 1. The similarity between *many-to-many* and *many-to-one* association type is 0.5. The similarity between *one-to-one* and anything else (*one-to-many* or *many-to-many*) is 0.

Feature 3 (value trend): Here, the sigmoid function defines similarity between 0 and 1.

$$Sim(f_3(c_1), f_3(c_2)) = \frac{1}{1 + e^{-x}}$$

where $f_3(c_i)$ is the *CoV* values (feature 3) for case c_i ; x computes how close *CoV* values for case 1 and case 2 are, defined as: $1/(|f_3(c_1) - f_3(c_2)|)$. Recall that *CoV* is the coefficient of variation as defined in Section 3.

The similarity between two cases is calculated based on the similarity of their feature values (the three introduced feature-based similarities above). Each feature-based similarity produces values between 0 and 1, and the total similarity function is the weighted sum of the three feature-based similarity (in which the weights are set to 1 in this work). Thus the total similarity is a value between 0 and 1.

5. EXPERIMENTS

We collected about 100 such use cases from IBM Watson Analytics and used 65% of them to be used as the known cases in our CBR system and the remaining 35% to evaluate our approach. Each case includes a question posed on a 2-dimensional spreadsheet of data in which we explicitly denote the *category* and *measure* columns, done manually by our domain experts. We then extracted the three features described in Section 3 for all the known cases in our CBR system. We then tested our approach using the remaining evaluation data.

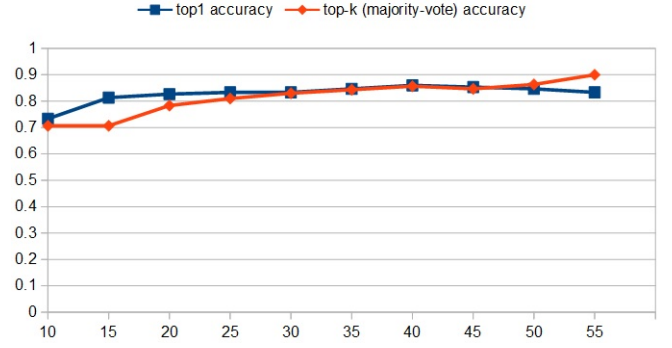


Figure 11: Trend of accuracy by increasing the number of training cases.

Our evaluations use a ‘majority voting’ scheme in which the top k most similar cases to the test case are selected and the aggregate action for the test case is selected based on the most frequent aggregate action of these k most similar cases. In the extreme, if $k = 1$, the action of the most similar ‘known’ case is picked as the aggregate action for the test case. For the majority vote, we empirically set $k = 3$, given that we find no significant difference in sweeping k from 3 to 6. Further increases of $k > 6$ results in lower accuracy.

Table 3 shows that the measure column’s *CoV* leads to the highest accuracy when each feature is used in isolation. Furthermore, when all the three features are used together, the approach achieves 86% accuracy using majority vote.

Table 3: The accuracy captured for the testing data using CBR and three different features.

Method	Accuracy
F1 (columns’ concept)	30%
F2 (columns’ association type)	30%
F3 (measure column’s <i>CoV</i>)	63%
All the three features	83%
All the three features with majority vote	86%

Furthermore, we calculate the accuracy trend over known cases. Figure 11 shows by increasing the known cases, while the testing data remains the same, our approach achieves better accuracy. Specifically, the accuracy increases from 70% to 90% by increasing the number of known cases from 10 to 55.

Notice that, in this work, we use the same feature weights for all the features (all feature weight are 1). In the future, we are going to learn feature weights using optimization techniques similar to those used by Lamontagne and Guyard [2014].

Finally, we calculate the system-level elapsed time in milliseconds for different tasks, shown in Table 4. The feature selection task (for all known cases) takes less than two minutes by a ThinkPad (T410s) with i5-2.4GHz and 8GB of RAM, and using Win 7 Professional 64bit.

The average time for learning aggregate actions of each test

Table 4: The system elapsed time in milliseconds for different kind of tasks.

Measure <i>CoV</i> values	Default Aggregation
Feature Extraction (known cases)	83594
Aggregate Learning (average)	8.46
Aggregate Learning (range)	$1 < t < 28$

case is about 8 milliseconds. The learning time varies for each test case (between 1 and 28 milliseconds) since the feature extraction for each test case is done at run time and, in particular, calculating the *CoV* highly depends on the number of columns in the data.

5.1 Examples of experimental results

Here we isolate three test cases and their most similar cases as identified by our approach, shown in Table 5. In the first two cases, our approach has been able to find the correct aggregate action, however our approach fails in the third case to learn the correct aggregate action.

Our approach does not learn the correct default aggregate action for test case 3 (code coverage of each component). The suggested action by our approach is *sum* instead of *last-period*. The code coverage spreadsheet keeps the percentage of code that has been tested over time for each component of a software. Thus, the correct aggregate action is taking *last-value*. We believe that this error is caused small number of rows in the code coverage spreadsheet. In particular, the calculated *CoV* for data in the code coverage dataset does not represent the trend of coverage values well enough.

6. CONCLUSION

In this work, we proposed a method for learning semi-additive behaviour in data analytics tools, chiefly IBM Watson Analytics in which these experiments were performed. Specifically, we introduced three features that are used in case-based reasoning for learning aggregate actions in business analytics. In particular, the data trend of a variable is an important factor that can be used for learning the default aggregate of that variable. We calculated the data trend using *coefficient of variation (CoV)*, whose use can empirically increase the accuracy up to 30%. Overall, our approach is able to learn the default aggregate of our testing data with up to 90% accuracy. In this work, we used the same feature weights for all the features (all feature weights are 1.0). In the future, we intend to learn feature weights using optimization techniques such as linear programming, and we are collecting more test cases to experiment our approach on larger data sets.

Acknowledgment

This research is supported in part by a contribution from IBM Canada.

References

- Chen, Y. Y., Ferrer, X., Wiratunga, N., and Plaza, E. (2014). Sentiment and preference guided social recommendation. In *Proceedings of 22nd International Conference on Case-Based Reasoning Research and Development, (ICCBR'14), Cork, Ireland*.
- Dong, R., O'Mahony, M. P., and Smyth, B. (2014). Further experiments in opinionated product recommendation. In *Proceedings of 22nd International Conference on Case-Based Reasoning Research and Development, (ICCBR'14), Cork, Ireland*.
- Freyne, J. and Smyth, B. (2010). Visualization for the masses: Learning from the experts. In *Case-Based Reasoning. Research and Development*, pages 111–125. Springer.
- Jalali, V. and Leake, D. (2014). On retention of adaptation rules. In *Proceedings of 22nd International Conference on Case-Based Reasoning Research and Development, (ICCBR'14), Cork, Ireland*.
- Jurisica, I. (1997). Similarity-based retrieval for diverse bookshelf software repository users. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research (CASCON'97), Toronto, ON, Canada*.
- Kolodner, J. L. (1993). *Case-based learning*, volume 10. Springer.
- Lamontagne, L. and Guyard, A. B. (2014). Learning case feature weights from relevance and ranking feedback. In *Proceedings of the 27th International Florida Artificial Intelligence Research Society Conference (FLAIRS'14), Pensacola Beach, Florida, USA*.
- Lamontagne, L. and Plaza, E., editors (2014). *Proceedings of 22nd International Conference on Case-Based Reasoning Research and Development, (ICCBR'14), Cork, Ireland*, volume 8765 of *Lecture Notes in Computer Science*. Springer.
- Rais-Ghasem, M., Grosset, R., Petitclerc, M., and Wei, Q. (2013). Towards semantic data analysis. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research (CASCON'13), Toronto, Ontario, Canada*.
- Riesbeck, C. K. and Schank, R. C. (1989). *Inside Case-Based Reasoning*. L. Erlbaum Associates Incorporation, Hillsdale, NJ, USA.

Table 5: Examples of our experimental data and results.

Test case 1	
Data set	American Time Use Survey.csv
Question	What are values of sleeping time over gender?
Category column	[Gender]
Measure column	[Sleeping]
Most similar known case	
Data set	GL Budget.xls
Question	What are values of budget over account?
category column	[Account-Number]
measure column	[Period-Budget-Amount]
aggregate action	[average]
Test case 2	
Data set	[Ticket-Sales-by-Section-with-Geo.csv]
Question	What are total tickets purchased for each event ?
Category column	[EventName]
Measure column	[Total Tickets Purchased]
Most similar known case	
Data set	[Ticket-Sales-by-Section-with-Geo.csv]
Question	What are total price of purchased tickets for each event ?
category column	[EventName]
measure column	[Total Ticket Purchase Price]
aggregate action	[sum]
Test case 3	
Data set	Sonar Results over Time.xls
Question	What is code coverage for each component?
Category column	[Component]
Measure column	[Coverage]
Most similar known case	
Data set	World Mortality.csv
Question	What are the number of deaths by cause?
category column	[Cause]
measure column	[Number of Deaths]
aggregate action	[sum]